# README.md

*Duncan Turnbull*

*21 December 2014*

## Overview

The project requirement is to take raw data and turn it into tidy data. The data is online in a zip file and is part of the "Human Activity Recognition Using Smartphones Data Set". It has come from a group of 30 volunteers aged between 19 & 48. The data is the accelerations recorded as they move using a smartphone accelerometer and gyroscope.

### Project Requirements

You should create one R script called run_analysis.R that does the following.

1. Merges the training and the test sets to create one data set.
2. Extracts only the measurements on the mean and standard deviation for each measurement.
3. Uses descriptive activity names to name the activities in the data set
4. Appropriately labels the data set with descriptive variable names.
5. From the data set in step 4, creates a second, independent tidy data set with the average of each variable for each activity and each subject.

### Part 1

- Download data and unzip
- load test and training sets
  - load row and column labels
    - get subset of labels that is mean and std before cleaning
    - clean column and row labels
    - apply labels
  - add variable to reflect whether test or train data
- Merge data

### Part 2: select std deviation and mean data only

- Once all data is in one table discard non necessary columns or
- could try reading in only required columns or remove straight away
  - would require an early logical vector of which columns are required.
  - whats the best way to do this?
  - looking for mean() and std() - could get caught by meanfreq() if dont use ()

### Part 3: Name the activities appropriately

- Need to define all included activities - walk, run, upstairs etc, these are the rows
- Add column called ActivityName which links from activity.txt

### Part 4

- Label variable names (columns)

### Part 5

- create a second, independent tidy data set with the average of each variable for each activity and each subject.

## Functional Overview

### Assumptions

The script assumes it is called from the "UCI HAR Dataset" directory. All its references to files are in this directory. The tidyoutput.txt will be stored in this directory.

In my setup this is what I use to check I am in the correct directory before calling the script.

```
> getwd()
[1] "E:/Users/duncan/R Programming/R DataCleaning/Data Cleaning Project/data/UCI HAR Dataset"
> source('E:/Users/duncan/R Programming/R DataCleaning/Data Cleaning Project/DataCleaningProject/run_ana
```

### Input Files

There are a range of files that need to be merged. They come from a Zip file. When the file is expanded out it creates a directory "UCI HAR Dataset" which contains the data files.

- The observations are raw data
  - "test/X_Test.txt" & "train/X_Train.txt"
  - these have 2947 and 7352 rows respectively and 561 variables in each
- Subject files providing subject ids for the observation rows
  - "test/subject_test.txt" & "train/subject_train.txt"
  - These range from 1 - 30
- Activity files providing the activity undertaken when the observation was made
  - the files are "train/ytrain.txt" & "test/ytest.txt"
  - and the readable description of the activities is in "activity_labels.txt"
  - The activities are:
    - WALKING
    - WALKING_UPSTAIRS
    - WALKING_DOWNSTAIRS
    - SITTING
    - STANDING
    - LAYING
- The different variables are in columns referenced by
  - "features.txt"
  - this provides the names for the columns read in from X_Test and X_Train
  - these are accelerations in different angles as well as mathmatical functions
- The "test/Inertial Signals" and "train/Inertial Signals"" folders are **NOT** referenced in this script

### Output files

The script creates one output file "tidyoutput.txt" in the directory it was called from

### The other files

The other files used are:

- README.md - this file which explains the workings of the script (hopefully)
- run_analysis.R - which is the actual script
- Codebook.Rmd - which explains the data in the output file

### Part 1

0. Setup parameters
    - Setup directories to access data folder
    - Assumptions:
        - we are starting in the working dir
        - we know specific files names and mapping to directories
1. Read labels and clean them
    - read the files "activity_labels.txt" ,"features.txt".
        - these are needed to label rows and set column variable name
    - Read the label files ( and clean them if needed )
    - Read the files into data tables and add vars for test/train (although it doesn't look like we need this anywhere)
    - apply the label from the files to the data
2. Read files

```
activitylabels <- read.table(file = "activity_labels.txt")
```

Get the required feature columns and note the ones we want for later than are mean() or std()

```
features <- read.table(file = "./features.txt")
wantedcols <- grep("(mean|std)\\(\\)",features$V2)
featurelabels <- gsub(features[,2],pattern = "[-(),]", replacement = ""  )
```

Get the activity data for both sets and join into one column

```
ytrain <- read.table(file = "./train//y_train.txt")
ytest <- read.table(file = "./test/y_test.txt")
activity <- rbind(ytest,ytrain)
```

Then convert the ids into names - for easier reading

```
activitynames <- as.character(activitylabels$V2)[activity$V1]
```

Create the Subject column from the two respective files

```
testsubject <- read.table(file = "./test//subject_test.txt")
trainsubject <- read.table(file = ".//train//subject_train.txt")
subject <- rbind(testsubject, trainsubject)
```

Get the big test set of data

```
xtest <- read.table(file = "./test/X_test.txt")
```

Set the names for the test data

```
colnames(xtest) <- features$V2
```

Get the big train set of data

```
xtrain <- read.table(file = "./train/X_train.txt")
```

Set the names for the train data

```
colnames(xtrain) <- features$V2
```

Add a variable type to show whether test or train data (not used now but just in case)

```
type <- character(nrow(xtest)+nrow(xtrain))
type[1:nrow(xtest)] <- "Test"
type[nrow(xtest)+1:nrow(xtrain)] <- "Train"
```

Create the biggest set of data by combining

```
data <- rbind(xtest,xtrain)
```

Capture the current names of the data

```
names <- names(data)
```

Join the data with the subject, activity and activity name ( for easier reading)

```
data <- cbind(subject,activity,activitynames,type,data)
```

Update the names for the new columns as they were messed up in the combine

```
colnames(data) <-c("Subject","Activity","ActivityName","Type", as.character(names))
```

Use previously selected active columns to remove anything but mean or std

```
data2 <-data[,c(1:4,wantedcols+4)]
```

Clean the variable names to get rid of non alpha characters

```
names2 <-names(data2)
names2 <- gsub(names2,pattern = "[-(),]", replacement = ""  )
names(data2) <- names2
```

**This gets us to Part 4 ( merged 3 parts together back there)**

Now we need to create a second independent data set with average of each variable for each activity and each subject:

- This seems like a good call to use dplyr group_by function to group by activity and subject and summarise to process everything
- What is needed is to group by activity and within that subject then create the average for each variable (column)
- The new data will have 6 activities X 30 subject = 180 rows and 66 averages per row

```
library(dplyr)

tidy <- group_by(data2, ActivityName, Subject)
tidyoutput <-summarise_each(tbl = tidy, funs(mean), c(-Type,-Activity))
```

Write out the tidy data

```
write.table( tidyoutput, row.name=FALSE ,file = "./tidyoutput.txt")
```

## Howto read the output file back in

To see the file you can use:

```
View( read.table(file="tidyoutput.txt", header=T) )
```

Note: the script will attempt to do this at the end anyway

## Tidy data

This output is tidy.

- One observation per row
- One variable per column
- The observational units form a table

Other clean up included: * Cleaned variable names of non alpha characters “(),_" * Only selection of required columns

## References:

- https://class.coursera.org/getdata-016/forum/thread?thread_id=100
- http://vita.had.co.nz/papers/tidy-data.pdf